APPENDIX A

C:\Documents and Settings\jhuggins\Local Settings\Temporary Internet Files\OLK4\Oper10/31/2001 6:00PM

```
//this is an example of intercepting an opengl call, and converting it into dual Direct3d8.
//This is one of the simplest examples possible.
//Some functions dont require much work at all.
//Other functions require extremly complex data conversion.
//this ClearDepth function, happens to be very similar to its d3d8 equavalent function
//  we know the val for depth is [0-1] which is same for input of d3d8's clear function.
//  thus no conversion of data required, just redirection.
//  If any conversion is required, it is done inside Opengl32.cpp.

//-----------------------------------------------------------------------------
//OPENGL32.CPP
//header for real function, written by SGI OpenGL
void (__stdcall* real_glClearDepth)(GLclampd depth);


//During init, we retrieve a pointer to the real opengl function
real_glClearDepth  = (void(__stdcall*)(GLclampd depth))GetProcAddress(DLLInst,"glClearDepth");



//inside our opengl32.dll wrapper, our pseudo function looks like this :
__declspec(dllexport) void __stdcall glClearDepth(GLclampd depth)
{
    if(convertTOd3d8)
    {//activly converting stream into d3d8dual
        //preform any necesary data conversion here.
        d3d_glClearDepth(depth);
    }
    else
    {//pass through, debug mode. normal OpenGL operation.
        real_glClearDepth(depth);
    }
}


//-----------------------------------------------------------------------------
//DUAL.CPP
//The opengl32.dll wrapper calls this function provided by our DualRendering System.
void d3d_glClearDepth(float depth)
{
    dual_glClearDepth(depth);
}

//the dual glClearDepth issues the commands to the 2 video cards.
void dual_glClearDepth(float depth)
{
    if(g_d3ddev1 != NULL)
    {
        g_d3ddev1->Clear(0,NULL,D3DCLEAR_ZBUFFER,D3DCOLOR_XRGB(0x00,0x00,0x00),depth,0);
    }
    if(g_d3ddev2 != NULL)
    {
        g_d3ddev2->Clear(0,NULL,D3DCLEAR_ZBUFFER,D3DCOLOR_XRGB(0x00,0x00,0x00),depth,0);
    }
}
```